Linux Network Servers

DNS – Parte 1

DNS é a abreviatura de Domain Name System. O DNS é um serviço de resolução de nomes. Toda comunicação entre os computadores e demais equipamentos de uma rede baseada no protocolo TCP/IP é feita através do número IP, porém não seria nada produtivo se os usuários tivessem que decorar/consultar uma tabela de números IP toda vez que tivessem que acessar um recurso da rede.

Por exemplo, você digita http://www.linux.com/, sem ter que se preocupar e nem saber qual o número IP do servidor onde está hospedado o site. Mas alguém tem que fazer este serviço, pois quando você digita http://www.linux.com, o protocolo TCP/IP precisa "descobrir" (o termo técnico é resolver o nome) qual o número IP está associado com o endereço digitado.

Se não for possível "descobrir" o número IP associado ao nome, não será possível acessar o recurso desejado. O papel do DNS é exatamente este, "descobrir", ou usando o termo técnico, "resolver" um determinado nome, como por exemplo www.linux.com. Resolver um nome significa, descobrir e retornar o número IP associado com o nome.

Em palavras mais simples, o DNS é um serviço de resolução de nomes, ou seja, quando o usuário tenta acessar um determinado recurso da rede usando o nome de um determinado servidor, é o DNS o responsável por localizar e retornar o número IP associado com o nome utilizado.

Durante os anos 70, Arpanet era uma pequena comunidade de algumas centenas de hosts. Um único arquivo, o HOSTS.TXT, continha toda a informação necessária sobre os hosts. Com o crescimento da ARPANET, entretanto, este esquema tornou-se inviável. O tamanho do arquivo HOST.TXT crescia na proporção em que crescia o número de hosts. Além disso, o tráfego gerado com o processo de atualização crescia em proporções ainda maiores uma vez que cada host que era incluído não só significava uma linha a mais no arquivo HOST.TXT, mas um outro host atualizando a partir do SRI-NIC.

Principais problemas que passaram a existir com o HOST.TXT:

* Tráfego e Carga: os problemas com tráfego na rede e carga do processador tornaram-se insuportáveis.

Linux Network Servers

* Nomes que coincidiam: Dois hosts do arquivo HOST.TXT não podiam ter o mesmo nome. Porém, apesar do NIC poder designar endereços únicos para cada host, ele não tinha nenhuma autoridade sobre os nomes dados aos mesmos.

* Consistência: Manter a consistência do arquivo com a rede se expandindo naquelas proporções se tornou cada vez mais difícil.

Para se resolver isso, um novo sistema de nomes deveria atender aos seguintes requisitos:

* Permitir que um administrador local tornasse os dados mundialmente disponíveis;

* Descentralização da administração para resolver o problema do gargalo gerado por um único host;

* O esquema deveria usar nomes em hierarquia para garantir a exclusividade dos nomes;

Queremos acessar o site www.linux.com. Precisamos resolver esse nome para um número IP. Podemos fazer isso simplesmente assim:

```
dig +short linux.com
216.34.181.51
```

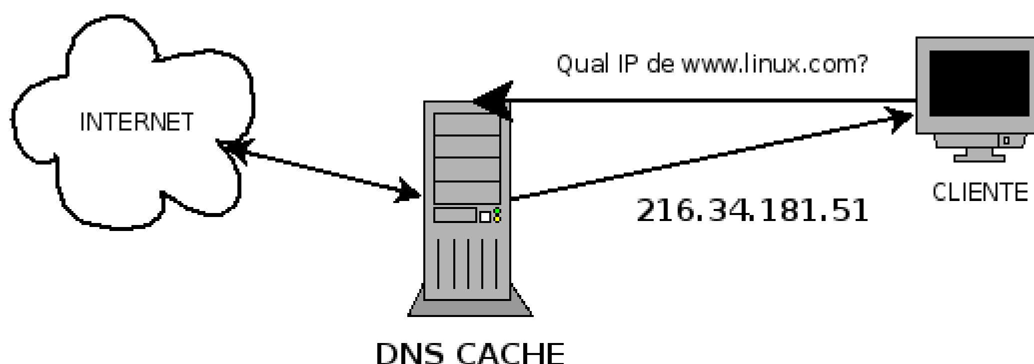
Se você colocar no seu navegador http://216.34.181.51 acessará o site corretamente.

Mas como o comando dig fez para conseguir o IP 216.34.181.51?

Nosso sistema faz as seguintes etapas: primeiro verifica se o www.linux.com existe no /etc/hosts. Se não, ele usará um dos name servers em /etc/resolv.conf e irá perguntar para eles. Agora que começa a ficar interessante.

Os IPs que colocamos no /etc/resolv.conf chamamos de servidores de cache ou simplesmente "resolvers" (resolvedores). Eles buscam os nomes na internet e armazenam uma cópia em memória (cache). A pergunta chega para o nosso DNS cache vindo de nossa máquina.

Linux Network Servers



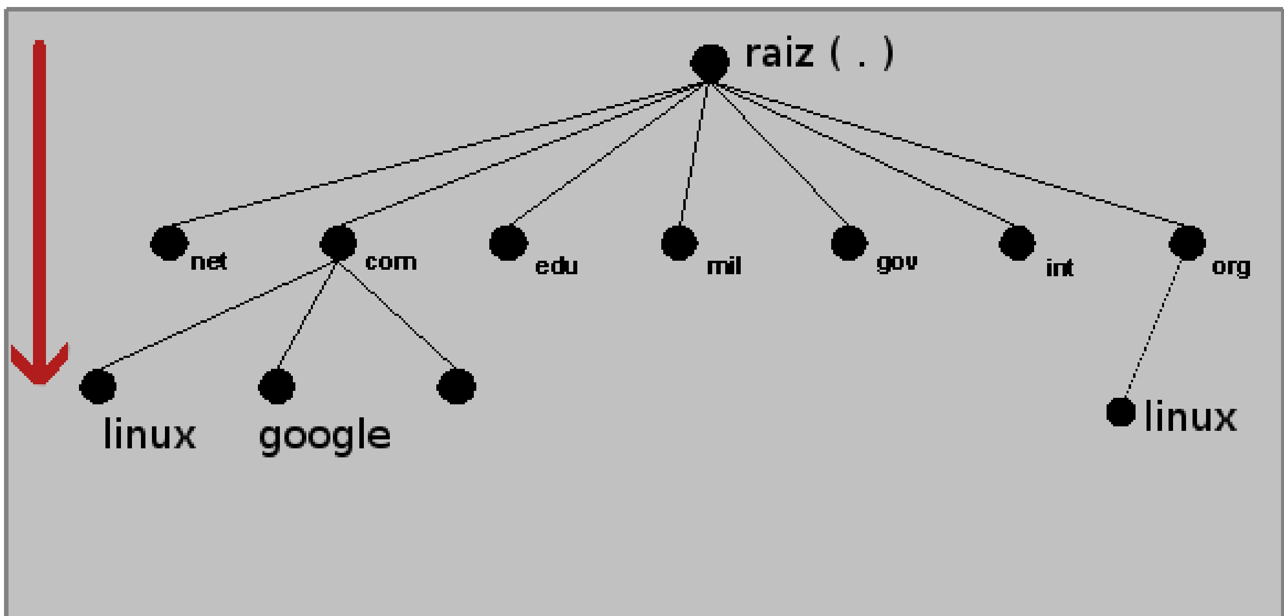
O DNS é hierárquico pois é baseado em conceitos tais como espaço de nomes e árvore de domínios. Assim existe isolamento de nomes e delegação de autoridade.

Nesta Figura é apresentada uma visão abreviada da estrutura do DNS definida para a Internet. O principal domínio, o domínio root, o domínio de mais alto nível foi nomeado como sendo um ponto (.). No segundo nível foram definidos os chamados "Top-level-domains" TLD. Estes domínios são bastante conhecidos, sendo os principais:

* com: Organizações comerciais
* gov: Organizações governamentais
* edu: Instituições educacionais
* org: Organizações não comerciais
* net: Serviços de rede e comunicação

O DNS cache recebe a pergunta do cliente por www.linux.com, ele não tem a resposta. Então inicia-se uma jornada para descobrir.

Linux Network Servers

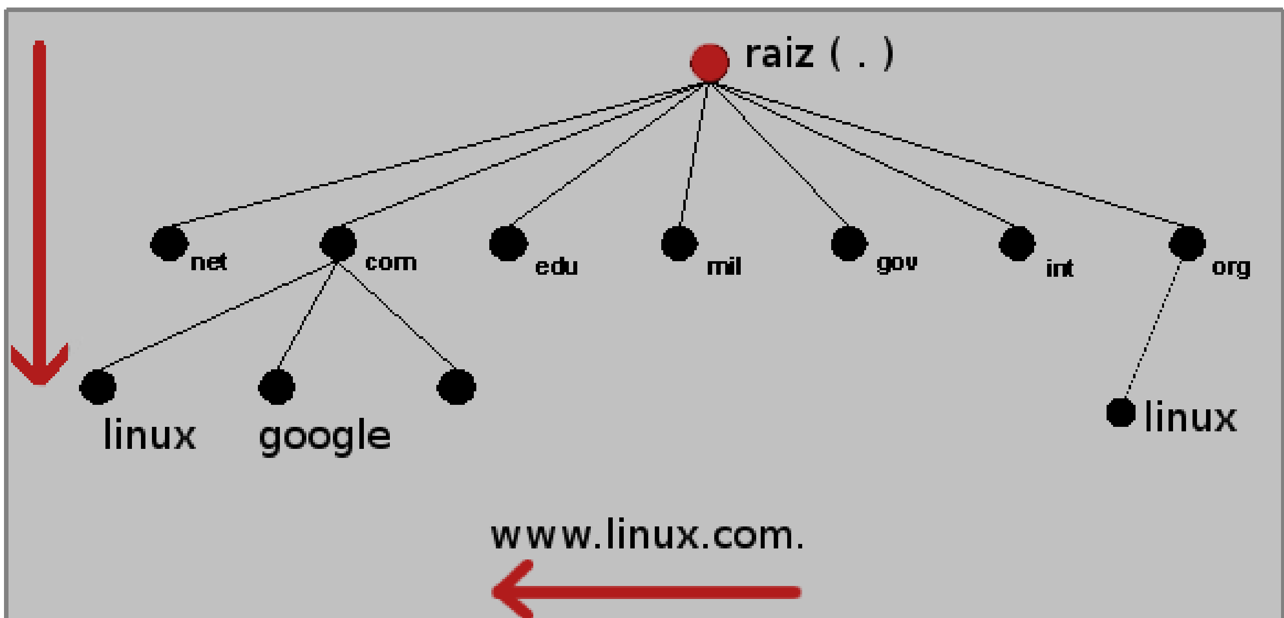


Devemos ler um endereço da direita para a esquerda. Todo endereço começa com um ponto. Coloque no seu navegador www.linux.com. <-- com um ponto no final mesmo, veja que funciona.

Bem, o DNS cache não sabe a resposta, então ele vai começar a procurar pela raíz. Os servidores da raiz são chamamos de ROOT servers.

Linux Network Servers

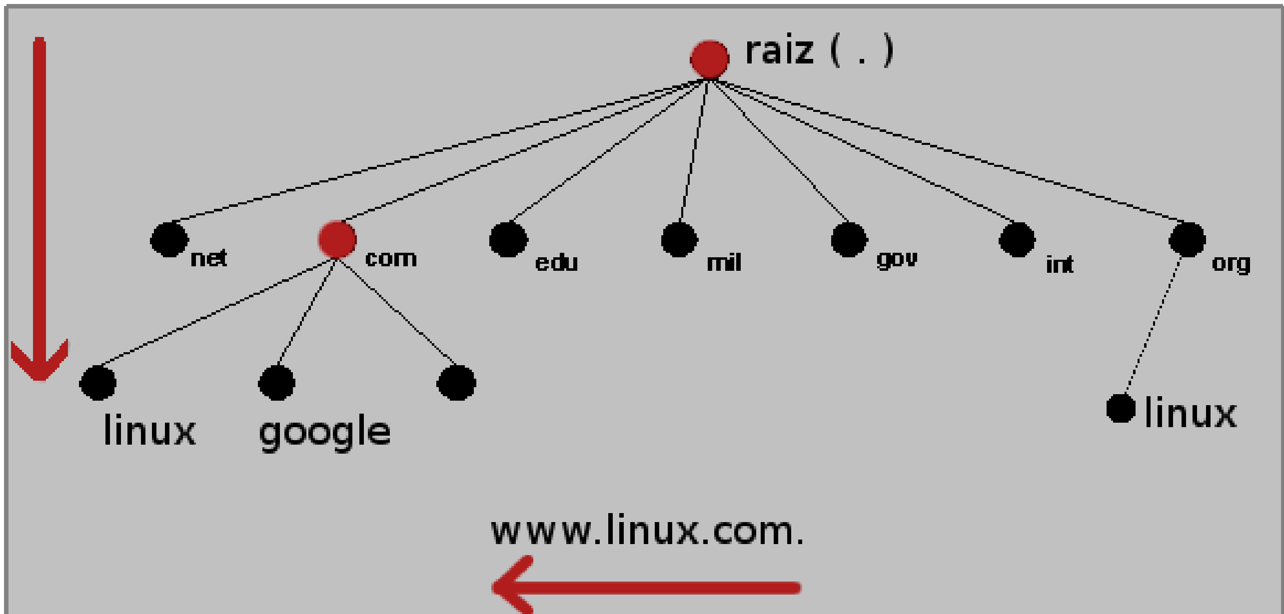
Então inicia-se a seguinte conversa:



* DNS cache: Oi root server, por acaso você sabe qual é o IP de www.linux.com.?

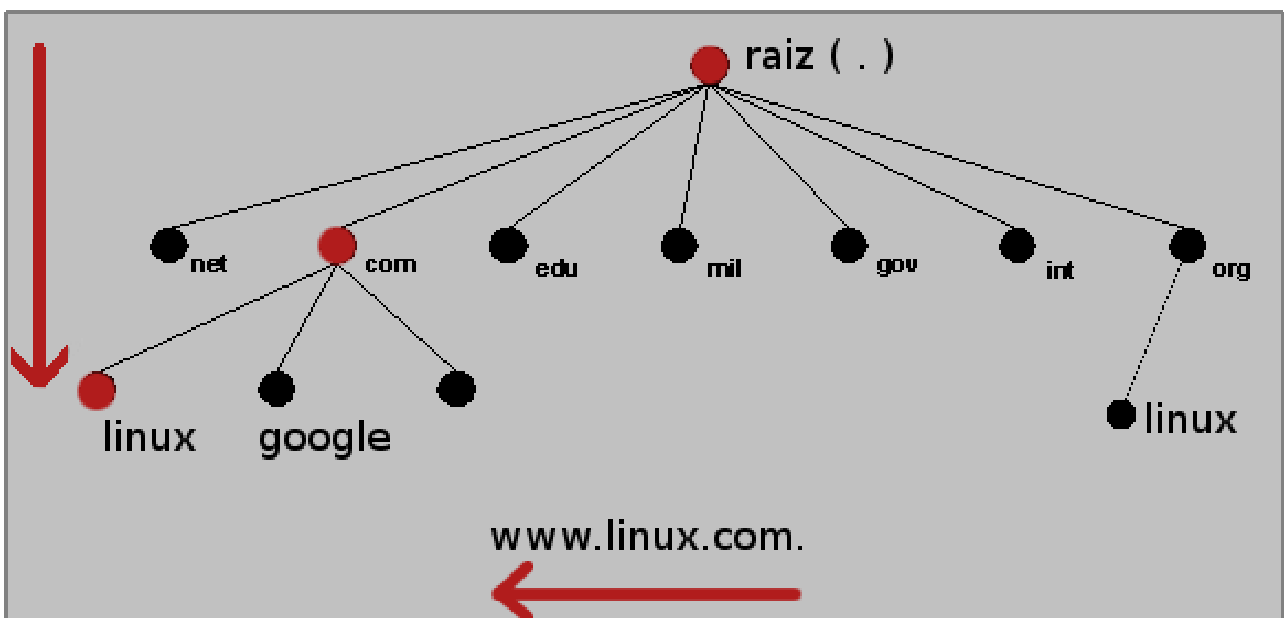
* ROOT SERVER: Não sei. Porém eu sei quem responde pelo .com, ele deve saber. Pergunte para ele.

Linux Network Servers



* DNS cache: Oi .com, você conhece www.linux.com.?

* .com: Não sei. Porém eu sei quem responde por linux.com., ele deve saber. Pergunte para ele.



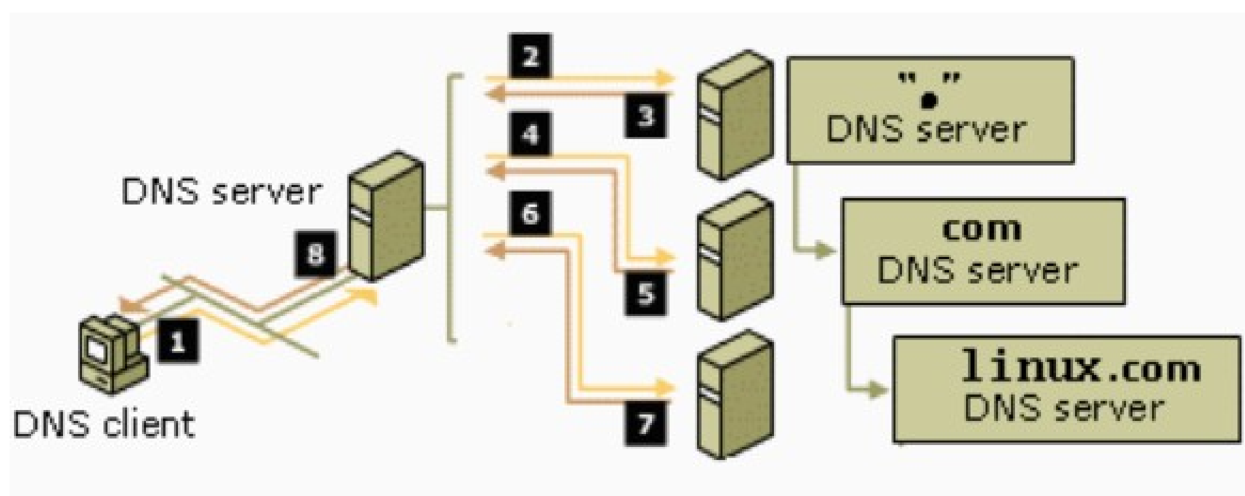
Linux Network Servers

* DNS cache: Oi linux.com., você conhece www.linux.com.?

* linux.com.: Sim, o IP é 216.34.181.51.

O DNS cache enfim obtém sua resposta, armazena-a e envia ao cliente. Isso é uma resolução recursiva.

A resposta que obtivemos é uma resposta autoritativa. O servidor que respondeu para o DNS cache chamamos de autoritativo.



Temos nossa resposta. A pergunta é: devemos guardá-la para sempre?
Não, existe o TTL. Endereços mudam.

Vamos também conhecer melhor o dig.

O comando dig é o acrônimo para "domain information groper", que significa algo como "aquele que busca por informações de domínio no escuro", e ao mesmo tempo, a palavra dig em inglês significa literalmente "escavar". Acho que mencionar estas curiosidades demonstra o esforço de imaginação dos criadores do dig, e não à toa, ele é o comando de pesquisa mais poderoso no pacote de utilitários BIND.

Linux Network Servers

Rode em dua máquina:

```
dig www.linux.com
```

```
miguel@ebl ~> dig www.linux.com

; <<>> DiG 9.2.9 <<>> www.linux.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57446
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.linux.com.                  IN      A

;; ANSWER SECTION:
www.linux.com.          3591    IN      CNAME   linux.com.
linux.com.              3591    IN      A       216.34.181.51

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Nov 30 00:33:49 2008
;; MSG SIZE  rcvd: 61
```

pergunta

Linux Network Servers

```
miguel@ebl ~> dig www.linux.com

; <<>> DiG 9.2.9 <<>> www.linux.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57446
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.linux.com.                 IN      A
                                                        resposta
;; ANSWER SECTION:                  TTL
www.linux.com.          3591    IN      CNAME   linux.com.
linux.com.              3591    IN      A       216.34.181.51

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Nov 30 00:33:49 2008
;; MSG SIZE  rcvd: 61
```

Resposta com TTL em segundos.

```
miguel@ebl ~> dig www.linux.com

; <<>> DiG 9.2.9 <<>> www.linux.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57446
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.linux.com.                 IN      A

;; ANSWER SECTION:
www.linux.com.          3591    IN      CNAME   linux.com.
linux.com.              3591    IN      A       216.34.181.51

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Nov 30 00:33:49 2008
;; MSG SIZE  rcvd: 61
miguel@ebl ~> dig www.linux.com

; <<>> DiG 9.2.9 <<>> www.linux.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57446
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.linux.com.                 IN      A

;; ANSWER SECTION:
www.linux.com.          3591    IN      CNAME   linux.com.
linux.com.              3591    IN      A       216.34.181.51

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Nov 30 00:33:49 2008
;; MSG SIZE  rcvd: 61
```

Linux Network Servers

São resource records, que faz parte do conteúdo da uma zona de um domínio. Traduzindo literalmente, um registro de recurso.

Isso é um domínio: linux.com.

Isso é um registro de recurso (resource record, ou RR): www

Zona de um domínio é um conjunto de resource records, como se fosse um banco de dados.

Tipos de Resource Records:

* Name Server - NS - Identifica o servidor de nomes de um domínio

```
dig -t ns hackerteen.com
```

* Address - A - Mapeia um hostname para um endereço
* Mail Exchanger - MX - Identifica o servidor de correio para um domínio

```
dig -t mx uol.com.br
```

* Canonical Name - CNAME - Define uma alias para um hostname

No exemplo anterior, o www.linux.com. é um CNAME para linux.com. E o IP de linux.com. é 216.34.181.51.

Conceitos que temos saber até aqui:

* Domínio: é um nome que serve para localizar e identificar conjuntos de computadores na Internet.

* Top Level Domain: primeiro domínio após a raíz.

Linux Network Servers

* Resource Record, ou RR: dado sobre um domínio, como por exemplo um host e seu respectivo IP.

* Zona: conjunto de resource records de um domínio.

* FQDN: O nome completo de um computador da rede é conhecido como FQDN – Full Qualifided Domain Name.

Por exemplo ftp.abc.com.br é um FQDN. ftp (a primeira parte do nome) é o nome de host e o restante representa o domínio DNS no qual está o computador. A união do nome de host com o nome de domínio é que forma o FQDN.

O BIND (Berkeley Internet Name Domain) é o servidor de nomes utilizado na grande maioria dos servidores da Internet, provendo uma estável e robusta arquitetura sobre a qual as organizações podem construir sua estrutura de nomes.

Instalar o BIND9 no Debian basta executar:

```
# aptitude install bind9
```

O arquivo de configuração principal do BIND9 chama-se named.conf, e nas distribuições Red Hat e Suse ele fica exatamente no diretório /etc.

No Debian, entretanto, este arquivo foi fragmentado em três. O arquivo principal ainda chama-se named.conf mas contém apenas configurações estáticas.

Ele utiliza a cláusula **include** para anexar os arquivos named.conf.options e named.conf.local.

Sendo que desses dois, o primeiro serve para personalizar todas opções referentes ao funcionamento do próprio BIND, enquanto que o segundo serve para declarar todas as zonas pelas quais este servidor deve responder.

Linux Network Servers

```
etch:/etc/bind# ls -l
total 44
-rw-r--r-- 1 root root  237 2008-07-06 23:07 db.0
-rw-r--r-- 1 root root  271 2008-07-06 23:07 db.127
-rw-r--r-- 1 root root  237 2008-07-06 23:07 db.255
-rw-r--r-- 1 root root  353 2008-07-06 23:07 db.empty
-rw-r--r-- 1 root root  256 2008-07-06 23:07 db.local
-rw-r--r-- 1 root root 1506 2008-07-06 23:07 db.root
-rw-r--r-- 1 root bind 1611 2008-07-06 23:07 named.conf
-rw-r--r-- 1 root bind  165 2008-07-06 23:07 named.conf.local
-rw-r--r-- 1 root bind  695 2008-07-06 23:07 named.conf.options
-rw-r----- 1 bind bind   77 2008-09-20 13:16 rndc.key
-rw-r--r-- 1 root root 1317 2008-07-06 23:07 zones.rfc1918
```

O arquivo db.root (/var/named/named.ca no RedHat) relaciona os endereços dos 13 servidores raiz, e é lido como zona hint, que será explicada adiante.

O BIND vai utilizar a porta 53/UDP para receber consultas, a porta 53/TCP para transferir zonas para servidores escravos, a porta 953/TCP para receber comandos via rndc (que dependem de chaves criptografadas), e portas udp altas podem ser dinamicamente atribuídas para efetuar consultas em outros servidores.

* Abra o arquivo /etc/bind/named.conf.local

```
vim /etc/bind/named.conf.local

zone “teste-ht.com.br” {
     type master;
     file “/etc/bind/db.teste-ht”;
};
```

Linux Network Servers

Coloque o seguinte conteúdo em db.teste-ht:

```
$TTL 86400
@       IN     SOA ns.teste-ht.com.br.          root.teste-ht.com.br.(
                       2008080901;          serial
                       8h;                  refresh
                       1h;                  retry
                       3d;                  expire
                       3d );                default_ttl

        IN     NS      ns.teste-ht.com.br.
        IN     MX      10 smtp.teste-ht.com.br.
ns      IN     A       192.168.0.1
smtp    IN     A       192.168.0.1
www     IN     A       192.168.0.1
```

A diretiva $TTL – Define a TTL default para registros de recurso que não especificam um tempo explícito para serem considerados válidos. O valor de TTL pode ser especificado como um número de segundos ou como uma combinação de números e letras.

Usando o formato alfanumérico, uma semana pode ser definido como: $TTL 1w

Os valores de letra que podem ser usados com o formato alfanumérico são:
w - para semana
d - para dia
h - para hora
m - para minuto
s - para segundo

Sobre o registro SOA, vão algumas explicações:

Todos os arquivos de zona começam com um registro SOA. O @ no campo de nome do registro SOA recorre à origem atual, que neste caso é ns.teste-ht.com.br.

O "IN" é abreviação de "Internet" e "SOA" de "Start of autority".

root.teste-ht.com.br. Indica um endereço de e-mail do administrador do DNS.

Linux Network Servers

Não é usado arroba (@) e sim um ponto normal.

2008080901 (serial) – é um número de série, um valor numérico que diz ao servidor escravo que o arquivo de zona foi atualizado. Para determinar se o arquivo foi alterado, o servidor escravo periodicamente consulta o registro SOA no servidor mestre. Se o número de série no registro SOA do servidor mestre for maior do que o número de série da cópia do servidor escravo da zona, o escravo transfere a zona inteira d o mestre.

O número de série da zona deveria ser aumentado toda vez que o domínio for atualizado, para manter os servidores escravos sincronizados com o mestre.

1h (refresh) - tempo que o servidor secundário vai aguardar até checar se há atualizações no servidor primário.

15m (retry) - em caso de falha do refresh, o tempo até a próxima verificação.

1w (expire) - O tempo que o secundário aguardará o primário voltar, se esgotar, o secundário para de responder por essa zona.

1h (negative caching TTL) - Se a zona expirar, esse será o tempo pelo qual um servidor cache armazenará a informação NXDOMAIN antes de iniciar uma nova busca recursiva. O máximo são 3 horas.

As linhas com “NS” (name server) indica quem são os servidores DNS responsáveis pelo domínio.

Um exemplo importante para entender a questão do MX:

```
@       IN MX    10 mail.seunome.com.br.
@       IN MX    30 outroserver.outroserver.com.br.
```

Referem aos servidores de e-mail.

MX significa “Mail Exchanger”. Essa linha é necessária quando você quer usar um servidor de e-mail.

Os números 10 e 30 indicam a prioridade de cada servidor. Quanto menor o número, maior a prioridade.

Linux Network Servers

* Feito isso, teste a sintaxe do arquivo de configuração:

```
cd /etc/bind
named-checkconf
```

* Teste o arquivo de zona:

```
named-checkzone teste-ht.com.br db.teste-ht
```

* Reinicie o BIND

```
/etc/init.d/bind9 restart
```

* Vamos testar, coloque no seu resolv.conf

```
nameserver 127.0.0.1
```

* E depois:

```
ping www.teste-ht.com.br
```

* Podemos usar o dig também:

```
dig @127.0.0.1 www.teste-ht.com.br
dig @127.0.0.1 -t mx teste-ht.com.br
```